UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/751,372	12/29/2000	Anthony X. Jarvis	00-BN-051 (STMI01-00051)	8275

30425 7590 09/18/2007

STMICROELECTRONICS, INC.
MAIL STATION 2346
1310 ELECTRONICS DRIVE
CARROLLTON, TX 75006

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
09/18/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/751,372
Filing Date: December 29, 2000
Appellant(s): JARVIS ET AL.

Daniel E. Venglarik (Reg. No. 39,409)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 07 June 2007 appealing from the Office action mailed 29 December 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,706,481	Hannah et al.	1-1998
5,761,469	Greenley	6-1998

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Greenley, U.S. Patent No. 5,761,469 (herein referred to as Greenley) in view of Hannah et al., U.S. Patent Number 5,706,481 (herein referred to as Hannah).

Regarding claims 1, 14, 23, and 29, taking claim 14 as exemplary, Greenley has taught a processing system comprising:

A data processor (Greenley 100 of Fig.1) comprising:

An instruction execution pipeline comprising N processing stages, each of said N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by said instruction execution pipeline (Greenley Col.1 lines 34-40);

A data cache (Greenley 180 of Fig.1) capable of storing data values used by said pending instruction (Greenley Col.1 lines 42-43);

A plurality of registers (150 of Fig.1) capable of receiving said data values from said data cache (Greenley Col.1 lines 41-45);

A load store unit (Greenley 130 of Fig.1) capable of transferring a first one of said data values from said data cache to a target one of said plurality of registers

during execution of a load operation (Greenley Col.1 lines 15-21, 63-67 and Col.2 lines 1-7, 13-15);

A shifter circuit (Greenley 160,170 of Fig.1) associated with said load store unit capable of one of a) shifting (Greenley Col.2 lines 19-31), b) sign extending (Greenley Col.2 lines 48-54), and c) zero extending (Greenley Col.2 lines 45-47) said first data value prior to loading said first data value into said target register;

A memory coupled to said data processor (Greenley Col.1 lines 41-43); and

A plurality of memory-mapped peripheral circuits coupled to said data processor for performing selected functions in association with said data processor (Greenley Col. 1 line 29 to Col. 2 line 7 and Col. 2 lines 16-31). In regards to Greenley, the ICACHE, prefetch unit, and memory subsystems perform selected functions, such as storing instructions, fetching instructions from selected locations, accessing words in memory, at and from certain locations from memory.

Greenley has not explicitly taught bypass circuitry associated with said load store unit capable of transferring said first data value from said data cache directly to said target register without processing said first data value in said shifter circuit. However, Greenley has taught a sign extension unit (Greenley 160 of Fig.1) that performs a function to fill in unoccupied bits of a register by extending its sign after it is loaded from the data cache but before it is stored in the register file (Greenley Col.2 lines 48-50). Hannah has taught bypassing functions circuitry capable of transferring said first data value to said target without processing said first data value in said shifter circuit (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14). A person of ordinary skill in the art at the time the invention was made would have

recognized that bypasses improve the performance of a system by minimizing delays from unnecessary functions (Hannah column 9, lines 38-44). Therefore, it would have been obvious to a person of ordinary skill in the art at the time was made to incorporate the bypassing of Hannah in the device of Greenley to improve system performance.

Claim 14 is nearly identical to claims 1, 23, and 29. Claim 1 differs from claim 14 in its lack of a main memory and memory-mapped peripheral circuits, but comprises the same data processor as claim 14, and is therefore rejected for the same reasons. Claim 23 differs in that it lacks the load store unit and pipeline limitations, but the rest of the limitations are similar to claim 14, and is therefore rejected for the same reasons. Claim 29 differs in that it lacks the load store unit and pipeline limitations but has memory and memory-mapped peripheral circuits, similar to claim 14, but the rest of the limitations are similar, and is therefore rejected for the same reasons.

Regarding claims 2 and 15, taking claim 15 as exemplary, Greenley in view of Hannah has taught the processing system as set forth in claim 14, wherein said bypass circuitry transfers said first data value from said data cache directly to said target register during a load word operation (see above rejection of claim 1). While Greenley has taught a different register size than the applicant (Greenley Col.2 lines 17-19), the situation when a register has no unoccupied bits after being loaded with data from a data cache remains the same, with the size of the register and word being moot. Therefore Greenley's loading of a double word has the same consequences as the applicant's loading of a word.

Claim 2 is nearly identical to claim 15. Claim 2 differs in its parent claim, but comprises the same data processor as claim 15, and is therefore rejected for the same reasons.

Regarding claims 3 and 16, taking claim 16 as exemplary, Greenley in view of Hannah has taught the data processor as set forth in claim 15, wherein said bypass circuitry (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14) transfers said first data value from said data cache directly to said target register at the end of two machine cycles (Greenley Col.4 lines 17-20).

Claim 3 is nearly identical to claim 16. Claim 3 differs in its parent claim, but comprises the same data processor as claim 16, and is therefore rejected for the same reasons.

Regarding claims 4 and 17, taking claim 17 as exemplary, Greenley in view of Hannah has taught the data processor as set forth in claim 14, wherein said shifter circuit one of a) shifts, b) sign extends, or c) zero extends said first data value prior to loading said first data value into said target register during a load half-word operation (see Greenley Col.2 lines 17-20, 24-30, 46-47).

Claim 4 is nearly identical to claim 17. Claim 4 differs in its parent claim, but comprises the same data processor as claim 17, and is therefore rejected for the same reasons.

Regarding claims 5 and 18, taking claim 18 as exemplary, Greenley in view of Hannah has taught the data processor as set forth in claim 17, wherein said shifter circuit loads said shifted first data value into said target register at the end of two machine cycles (Greenley Col.4 lines 17-20), but has not explicitly taught the load taking three machine cycles. However, Greenley has taught this two-cycle latency for all load instructions, including those without the need for sign extension. In the situation where the load instruction fills the target register completely and no sign extension is needed, such as when the data is already properly aligned since it is coming from the data cache, which only contains aligned data, or from the ALU,

which outputs aligned data, the data processor as configured above will execute the load instruction at least one cycle faster due to the elimination of the shifter operations. This will create a latency of at least one cycle fewer for those load instructions which bypass the sign extension unit, and at least one more cycle for those which need sign extension, such as half-word load instructions. Because the applicant's claimed load instructions, which take 2 and 3 cycles for bypassed and sign-extended instructions, respectively, have no claimed advantages over the 1 and 2 cycles that Greenly in view of Hannah have taught, but are merely a change in the magnitude of latency, they are considered to be equivalent and thus taught by Greenley in view of Hannah (see *In re Rose*, 220 F.2d 459, 463, 105 USPQ 237, 240 (CCPA 1955)).

Claim 5 is nearly identical to claim 18. Claim 5 differs in its parent claim, but comprises the same data processor as claim 18, and is therefore rejected for the same reasons.

Regarding claims 6 and 19, taking claim 19 as exemplary, Greenley in view of Hannah has taught the data processor as set forth in claim 14, wherein said shifter circuit one of a) shifts, b) sign extends, and c) zero extends said first data value prior to loading said first data value into said target register during a load byte operation (see Greenley Col.2 lines 17-20, 36-40, 46-47).

Claim 6 is nearly identical to claim 19. Claim 6 differs in its parent claim, but comprises the same data processor as claim 19, and is therefore rejected for the same reasons.

Regarding claims 7 and 20, taking claim 20 as exemplary, Greenley in view of Hannah has taught the data processor as set forth in claim 6, wherein said shifter circuit loads said shifted first data value into said target register at the end of two machine cycles (Greenley Col.4 lines 17-20), but has not explicitly taught the transfer taking three machine cycles. However, Greenley has taught this two-cycle latency for all load instructions, including those without the

need for sign extension. In the situation where the load instruction fills the target register completely and no sign extension is needed, such as when the data is already properly aligned since it is coming from the data cache, which only contains aligned data, or from the ALU, which outputs aligned data, the data processor as configured above will execute the load instruction at least one cycle faster due to the elimination of the shifter operations. This will create a latency of at least one cycle fewer for those load instructions which bypass the sign extension unit, and at least one more cycle for those which need sign extension, such as half-word load instructions. Because the applicant's claimed load instructions, which take 2 and 3 cycles for bypassed and sign-extended instructions, respectively, have no claimed advantages over the 1 and 2 cycles that Greenly in view of Hannah have taught, but are merely a change in the magnitude of latency, they are considered to be equivalent and thus taught by Greenley in view of Hannah (see *In re Rose*, 220 F.2d 459, 463, 105 USPQ 237, 240 (CCPA 1955)).

Claim 7 is nearly identical to claim 20. Claim 7 differs in its parent claim, but comprises the same data processor as claim 20, and is therefore rejected for the same reasons.

Regarding claims 8, 9, 21, and 22, taking claims 21 and 22 as exemplary, Greenley in view of Hannah has taught the data processor as set forth in claim 14, but Greenley has not explicitly taught

> Wherein said bypass circuitry comprises a multiplexer having a first input channel coupled to a data output of said data cache; and
>
> Wherein said multiplexer has a second input channel coupled to an output of said shifter circuit.

However, Greenley has taught a sign extension unit (Greenley 160 of Fig.1) that fills in unoccupied bits of a register by extending its sign after it is loaded from the data cache but before reaching a register in the register file (Greenley Col.2 lines 48-50). Hannah has taught

> Wherein said bypass circuitry comprises a multiplexer having a first input channel (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14); and
>
> Wherein said multiplexer has a second input channel coupled to an output of another device (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14).

A person of ordinary skill in the art at the time the invention was made would have recognized that bypasses improve the performance of a system by minimizing delays from unnecessary functions (Hannah column 9, lines 38-44). Therefore, it would have been obvious to a person of ordinary skill in the art at the time was made to incorporate the bypassing of Hannah in the device of Greenley to improve system performance.

Claims 8 and 9 are nearly identical to claims 21 and 22 respectively. Claims 8 and 9 differ in its parent claim, but comprises the same data processor as claims 21 and 22, and is therefore rejected for the same reasons.

Regarding claim 10, Greenley has taught for use in a processor comprising an N-stage execution pipeline (Greenley Col.1 lines 34-40), a data cache (Greenley 180 of Fig.1), and a plurality of registers (Greenley 150 of Fig.1), a method of loading a first data value from the data cache into a target one of the registers, the method comprising the steps of:

Determining if a pending instruction in the execution pipeline is one of a load word operation, a load half-word operation, and a load byte operation (Greenley Col.1 lines 63-67, Col.2 lines 1-7, 17-19 and Col.5 lines 13-24).

In response to a determination that the pending instruction is a load half-word operation, transferring the first data value from the data cache to a shifter circuit and shifting the first data value prior to loading the first data value into the target register (Greenley Col.2 lines 24-31).

In response to a determination that the pending instruction is a load byte operation, transferring the first data value from the data cache to a shifter circuit and shifting the first data value prior to loading the first data value into the target register (Greenley Col.2 lines 35-40).

Greenley has not explicitly taught where in response to a determination that the pending instruction is a load word operation, transferring the first data value from the data cache directly to the target register without processing the first data value in the shifter circuit. However, Greenley has taught a sign extension unit (Greenley 160 of Fig.1) that fills in unoccupied bits of a register by extending its sign after it is loaded from the data cache but before it is stored in a certain register in the register file (Greenley Col.2 lines 48-50). Hannah has taught bypassing sign extension (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14). A person of ordinary skill in the art at the time the invention was made would have recognized that bypasses improve the performance of a system by minimizing delays from unnecessary functions (Hannah column 9, lines 38-44). Therefore, it would have been obvious to a person of

ordinary skill in the art at the time was made to incorporate the bypassing of Hannah in the device of Greenley to improve system performance.

Regarding claim 11, Greenley in view of Hannah has taught the method as set forth in claim 10, wherein the step of transferring the first data value requires two machine cycles during a load word operation (Greenley Col.4 lines 17-20). While Greenley has taught a different register size than the applicant (Greenley Col.2 lines 17-19), the situation when a register has no unoccupied bits after being loaded with data from a data cache remains the same, with the size of the register and word being moot. Therefore Greenley's loading of a double word has the same consequences as the applicant's loading of a word (*In re Rose*, 220 F.2d 459, 463, 105 USPQ 237, 240 (CCPA 1955)).

Regarding claim 12, Greenley in view of Hannah has taught the method as set forth in claim 10, wherein the step of transferring the first data value requires two machine cycles during a load half-word operation (Greenley Col.4 lines 17-20), but has not explicitly taught the transfer taking three machine cycles. However, Greenley has taught this two-cycle latency for all load instructions, including those without the need for sign extension. In the situation where the load instruction fills the target register completely and no sign extension is needed, such as when the data is already properly aligned since it is coming from the data cache, which only contains aligned data, or from the ALU, which outputs aligned data, the data processor as configured above will execute the load instruction at least one cycle faster due to the elimination of the shifter operations. This will create a latency of at least one cycle fewer for those load instructions which bypass the sign extension unit, and at least one more cycle for those which need sign extension, such as half-word load instructions. Because the applicant's claimed load

instructions, which take 2 and 3 cycles for bypassed and sign-extended instructions, respectively, have no claimed advantages over the 1 and 2 cycles that Greenly in view of Hannah have taught, but are merely a change in the magnitude of latency, they are considered to be equivalent and thus taught by Greenley in view of Hannah (see *In re Rose*, 220 F.2d 459, 463, 105 USPQ 237, 240 (CCPA 1955)).

Regarding claim 13, Greenley in view of Hannah has taught the method as set forth in claim 10 wherein the step of transferring the first data value requires two machine cycles during a load byte operation (Greenley Col.4 lines 17-20), but has not explicitly taught the transfer taking three machine cycles. However, Greenley has taught this two-cycle latency for all load instructions, including those without the need for sign extension. In the situation where the load instruction fills the target register completely and no sign extension is needed, such as when the data is already properly aligned since it is coming from the data cache, which only contains aligned data, or from the ALU, which outputs aligned data, the data processor as configured above will execute the load instruction at least one cycle faster due to the elimination of the shifter operations. This will create a latency of at least one cycle fewer for those load instructions which bypass the sign extension unit, and at least one more cycle for those which need sign extension, such as half-word load instructions. Because the applicant's claimed load instructions, which take 2 and 3 cycles for bypassed and sign-extended instructions, respectively, have no claimed advantages over the 1 and 2 cycles that Greenly in view of Hannah have taught, but are merely a change in the magnitude of latency, they are considered to be equivalent and thus taught by Greenley in view of Hannah (see *In re Rose*, 220 F.2d 459, 463, 105 USPQ 237, 240 (CCPA 1955)).

Referring to claim 24, Greenley in view of Hannah has taught

The data value is transferred from the cache to the target register via the bypass circuit (see above rejection of claim 1). While Greenley has taught a different register size than the applicant (Greenley Col.2 lines 17-19), the situation when a register has no unoccupied bits after being loaded with data from a data cache remains the same, with the size of the register and word being moot. Therefore Greenley's loading of a double word has the same consequences as the applicant's loading of a word.

The data value is transferred from the cache to the target register via the shifter circuit during a load half-word operation or a load byte operation (see Greenley Col.2 lines 17-20, 24-30, 46-47).

Referring to claim 25, Greenley in view of Hannah has taught

The bypass circuit (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14) is capable of transferring the data value from the cache to the target register at an end of two machine cycles (Greenley Col.4 lines 17-20); and

The shifter circuit is capable of providing the modified data value to the target register at an end of three machine cycles (Greenley Col.4 lines 17-20), but has not explicitly taught the load taking three machine cycles. However, Greenley has taught this two-cycle latency for all load instructions, including those without the need for sign extension. In the situation where the load instruction fills the target register completely and no sign extension is needed, such as when the data is already properly aligned since it is coming from the data cache, which only contains aligned data, or from the ALU, which outputs aligned data, the data processor as configured

above will execute the load instruction at least one cycle faster due to the elimination of the shifter operations. This will create a latency of at least one cycle fewer for those load instructions which bypass the sign extension unit, and at least one more cycle for those which need sign extension, such as half-word load instructions. Because the applicant's claimed load instructions, which take 2 and 3 cycles for bypassed and sign-extended instructions, respectively, have no claimed advantages over the 1 and 2 cycles that Greenly in view of Hannah have taught, but are merely a change in the magnitude of latency, they are considered to be equivalent and thus taught by Greenley in view of Hannah (see *In re Rose*, 220 F.2d 459, 463, 105 USPQ 237, 240 (CCPA 1955)).

Referring to claim 26, Greenley in view of Hannah has taught wherein the bypass circuit comprises a multiplexer having a first input coupled to the cache and a second input coupled to the shifter circuit (Hannah column 9, lines 31-67; Figure 11; Figure 12; Figure 13; and Figure 14).

Referring to claim 27, Greenley in view of Hannah has taught

> Shifting, sign extending, or zero extending a first data value from a cache and providing a modified first data value to a first of a plurality of registers (Greenley Col.1 lines 15-21, 63-67 and Col.2 lines 1-7, 13-15, and 19-54); and
>
> Transferring a second data value from the cache to a second of the plurality of registers without shifting, sign extending, or zero extending the second data value (Greenley Col.1 lines 15-21, 63-67 and Col.2 lines 1-7, 13-15, and 19-54).

Referring to claim 28, Greenley in view of Hannah has taught

Shifting, sign extending, or zero extending the first data value comprises shifting, sign extending, or zero extending the first data value in response to determining that a first pending instruction in a processor is a load byte operation or a load half-word operation (see Greenley Col.2 lines 17-20, 24-30, 46-47); and

Transferring the second data value comprises transferring the second data value to the second register in response to determining that a second pending instruction in the processor is a load word operation (see Greenley Col.2 lines 17-20, 24-30, 46-47).

The table below shows a more detailed breakdown of rejection of independent claim 14 above.

Instant Application	Prior Art
A data processor comprising:	Greenley Figure 1, element 100
An instruction execution pipeline comprising N processing stages, each of said N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by said instruction execution pipeline;	Greenley Column 1, lines 34-40 "...Pipelining allows the processing of instructions to occur in stages..."
A data cache capable of storing data values used by said pending instruction;	Greenley Figure 1, element 180 Column 1, lines 42-43 "...data cache **180**..."
A plurality of registers capable of receiving said data values from said data cache;	Greenley Figure 1, element 150 Column 1, lines 41-45 "...register file **150**..."
A load store unit capable of transferring a first one of said data values from said data cache to a target one of said plurality of registers during execution of a load operation;	Greenley Figure 1, element 130 Column 1, lines 15-21 "Processors use LOAD instructions to fetch data from memory..."

	Column 1, line 63 to column 2, line 7 "...If the one or more of the instructions is a LOAD instruction..." Column 2, lines 13-15 "...the needed data is fetched from the data cache **180**..."
A shifter circuit associated with said load store unit capable of one of a) shifting, b) sign extending, and c) zero extending said first data value prior to loading said first data value into said target register;	Greenley Figure 1, element 160 Figure 1, element 170 Column 2, lines 19-31 "...the alignment unit **170**..." Column 2, lines 45-54 "...The purpose of the sign extension unit **160**..."
Bypass circuitry associated with said load store unit capable of transferring said first data value from said data cache directly to said target register without processing said first data value in said shifter circuit;	Hannah Figure 11 Figure 12 Figure 13 Figure 14 Column 9, lines 31-67 "...A number of multiplexers are programmed to couple the various units in order to achieve the desired configuration...these capabilities are disabled or bypassed as appropriate for the position of the slice in the larger word..." Greenley has taught a sign extension unit (Greenley 160 of Fig.1) that performs a function to fill in unoccupied bits of a register by extending its sign after it is loaded from the data cache but before it is stored in the register file (Greenley Col.2 lines 48-50 "...the data loaded from the data cache **180**..."). In regards to Hannah, the interpolator chains of multiplexers allows for the system to bypass the sign extension depending on the location of the most and least significant bits in the data word, i.e. whether the most significant bits are correctly aligned in the data word. When the most and least significant bits are in the correct positions, the sign extension is bypassed, otherwise the sign extension is

	performed (Hannah column 9, lines 31-67 "...Since each slice handles both a sign bit and its extension, as well as fraction bits, these capabilities are disables or bypassed as appropriate for the position of the slice in the larger word..."). A person of ordinary skill in the art at the time the invention was made would have recognized that bypasses improve the performance of a system by minimizing delays from unnecessary functions (Hannah column 9, lines 38-44 "...the same hardware can be utilized with minimal extra gate count to attain the flexibility in data width...in other applications, speed and cost are of greater importance."). Therefore, it would have been obvious to a person of ordinary skill in the art at the time was made to incorporate the bypassing of Hannah in the device of Greenley to improve system performance.
A memory coupled to said data processor; and	Greenley Col.1 lines 41-43
A plurality of memory-mapped peripheral circuits coupled to said data processor for performing selected functions in association with said data processor.	Greenley Col. 1 line 29 to Col. 2 line 7 and Col. 2 lines 16-31 In regards to Greenley, the ICACHE, prefetch unit, and memory subsystems perform selected functions, such as storing instructions, fetching instructions from selected locations, accessing words in memory, at and from certain locations from memory.

The table below shows how independent claims 1, 23, 27, and 29 have similar limitations to claim 14, which are rejected for the same reasons laid forth in the above.

Claim 1	Claim 14	Claim 23	Claim 27	Claim 29
A data processor comprising:	A data processor comprising:	A processor, comprising:	A method, comprising:	A system, comprising: A processor comprising:

An instruction execution pipeline comprising N processing stages, each of said N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by said instruction execution pipeline;	An instruction execution pipeline comprising N processing stages, each of said N processing stages capable of performing one of a plurality of execution steps associated with a pending instruction being executed by said instruction execution pipeline;			
A data cache capable of storing data values used by said pending instruction;	A data cache capable of storing data values used by said pending instruction;	A cache;		A cache;
A plurality of registers capable of receiving said data values from said data cache;	A plurality of registers capable of receiving said data values from said data cache;	A plurality of registers;		A plurality of registers;
A load store unit capable of transferring a first one of said data values from said data cache to a target one of said plurality of registers during execution of a load operation;	A load store unit capable of transferring a first one of said data values from said data cache to a target one of said plurality of registers during execution of a load operation;			
A shifter circuit associated with said load store	A shifter circuit associated with said load store	A shifter circuit capable of shifting, sign	Shifting, sign extending,	A shifter circuit capable of shifting,

unit capable of one of a) shifting, b) sign extending, and c) zero extending said first data value prior to loading said first data value into said target register; and	unit capable of one of a) shifting, b) sign extending, and c) zero extending said first data value prior to loading said first data value into said target register;	extending, or zero extending a data value from the cache and providing a modified data value to a target one of the registers; and	or zero extending a first data value from a cache and providing a modified first data value to a first of a plurality of registers	sign extending, or zero extending a data value from the cache and providing a modified data value to a target one of the registers; and
Bypass circuitry associated with said load store unit capable of transferring said first data value from said data cache directly to said target register without processing said first data value in said shifter circuit.	Bypass circuitry associated with said load store unit capable of transferring said first data value from said data cache directly to said target register without processing said first data value in said shifter circuit;	A bypass circuit capable of transferring the data value from the cache to the target register without processing the data value in the shifter circuit.	Transferring a second data value from the cache to a second of the plurality of registers without shifting, sign extending, or zero extending the second data value	A bypass circuit capable of transferring the data value from the cache to the target register without processing the data value in the shifter circuit;
	A memory coupled to said data processor; and			A memory coupled to said data processor; and
	A plurality of memory-mapped peripheral circuits coupled to said data processor for performing selected functions in association with said data processor.			A plurality of memory-mapped peripheral circuits coupled to said data processor for performing selected functions in association with said data processor.

(10) Response to Argument

Applicant argues in essence on pages 12-63

> ...It is clear from this that, in Greenley's system, that the 'shifter circuit' of 160/170 cannot be bypassed, as those functions must be present...
>
> ...
>
> ...Hanna (*sic*) does not include any bypass circuitry that transfers any data values from a data cache to a register...

This has not been found persuasive. Applicant's arguments rely upon Greenley column 2, lines 19-30, specifically the statement

> Since needed data may not be physically stored consecutively in data cache **180**, a LOAD access to data cache **180** must insure that the accessed data is aligned (i.e. both the upper and lower half of a word is fetched) into an appropriate format to write into a register in the register file **150** (Greenley column 2, lines 19-25)...

Applicants' contend that this statement means that "the alignment must be assured, and that alignment unit 170 assures that the appropriate bits of the half-words are in the appropriate bit positions (Arguments page 18, lines 18-19)." Applicants' further use this statement as a basis to further insinuate that Hannah defeats this purpose, i.e. does not assure alignment, by bypassing the alignment unit. However, Hannah does not defeat that purpose. The Examiner agrees that Greenley requires the data to be aligned, i.e. "the alignment must be assured", and that the alignment unit assures the data is aligned, but the alignment unit is only needed when the "data may not be physically stored consecutively in data cache (Greenley column 2, line 19)". Hannah clearly discloses that their sign extension unit performs alignment, similar to Greenley, but the

system can bypass the sign extension depending on the location of the most and least significant bits in the data word, i.e. whether the most and least significant bits are correctly aligned in the data word. When the most and least significant bits are in the correct positions, i.e. when the significant bits are aligned properly, the sign extension is bypassed, otherwise the sign extension is performed (Hannah column 9, lines 31-67 "...Since each slice handles both a sign bit and its extension, as well as fraction bits, these capabilities are disables or bypassed as appropriate for the position of the slice in the larger word..."). Applicants' arguments insinuate that Greenley suggests that data must be processed by the sign extension unit and aligning unit all the time, even when the data is aligned. However, Greenley clearly states that processing through the sign extension unit and the aligning unit is only necessary when the data is not aligned properly. Similarly, Hannah has a sign extension and alignment unit for aligning data when the data is out of alignment, as required by Greenley, but further discloses that the sign extension and alignment unit is bypassed only when the data is already aligned, which makes the system faster since unneeded elements and functions are skipped.

Applicant argues in essence on pages 27, 34-35, and 42-43

> ...the Examiner hypothesizes what latency savings might occur in the proposed combination, without making any reference to the teachings of the art itself, and indicates that the claim limitation is simply a 'change in magnitude of latency'...

This has not been found persuasive. This argument appears to be based upon the invalidity of the combination of Greenley and Hannah, but, as argued above, the combination is valid. As explained in the rejection above, the LOAD instruction takes two cycles, as stated by Greenley in column 4, lines 17-20. Greenley has taught this two-cycle latency for all load

instructions, including those without the need for sign extension. In the situation where the load instruction fills the target register completely and no sign extension is needed, such as when the data is already properly aligned since it is coming from the data cache, which only contains aligned data, or from the ALU, which outputs aligned data, the data processor as configured above will execute the load instruction at least one cycle faster due to the elimination of the shifter operations. This will create a latency of at least one cycle fewer for those load instructions which bypass the sign extension unit, and at least one more cycle for those which need sign extension, such as half-word load instructions. The applicant's claimed load instructions, which take 2 and 3 cycles for bypassed and sign-extended instructions, respectively, have no claimed advantages or claimed significant structural and/or functional differences over the 1 and 2 cycles of Greenly and Hannah. As such, the choice between 1 and 2 cycles of Greenley and Hannah or the 2 and 3 cycles of the claims appears to be an arbitrary, numerical change in magnitude for the latency, which is not patentable subject matter.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Aimee J. Li
Patent Examiner
AU 2183

Conferees:

Eddie Chan
Supervisory Patent Examiner
AU 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

/Lynne H Browne/
Lynne H Browne
Appeal Practice Specialist, TQAS
Technology Center 2100